What is claimed is:

Sub A2

1. A microprocessor system for executing instructions described in a program comprising:

a main processor for executing by means of hardware those instructions which belong to a first instruction set and for executing by means of software those instructions which belong to a second instruction set; and

a co-processor operative under the control of said main processor for autonomously fetching an instruction belonging to said second instruction set to execute same by means of its hardware.

2. A microprocessor system according to claim 1, wherein said co-processor detects an encounter with a specific one of the instructions belonging to said second instruction set which said co-processor cannot process by itself and issues a notification of said encounter to said main processor to thereby request the main processor to execute said specific instruction.

3. A microprocessor system according to claim 2, wherein said co-processor detects an encounter with a specific one of the instructions belonging to said second instruction set for which data present under the control of said main processor needs to be handled to thereby determine that said co-processor has encountered a specific instruction which cannot be processed by itself.

4. A microprocessor system according to claim 2, wherein said co-processor issues said notification by means of dedicated interrupt vectors assigned in advance

respectively to a predetermined number of ones of the instructions belonging to said second instruction set which have a higher frequency of execution than the other instructions.

5. A microprocessor system according to claim 4, wherein each of said dedicated interrupt vectors is assigned to a plurality of instructions belonging to said second instruction set.

6. A microprocessor system according to claim 4, wherein priorities are set to a plurality of ones of said dedicated interrupt vectors.

7. A microprocessor system according to claim 6, wherein a single instruction is assigned to a given one of said dedicated interrupt vectors to which a higher priority is set, while a plurality of instructions are assigned to a given one of said dedicated interrupt vectors to which a lower priority is set.

8. A microprocessor system according to claim 2, wherein said co-processor further comprising:

a stack memory for holding data generated in the course of execution of an instruction which belongs to said second instruction set;

a stack pointer for holding an address of the most recent data in said stack memory; and

a hardware resource for carrying out a process for updating said stack pointer among processes which take place in the course of execution of said specific instruction.

A2

9. A microprocessor system according to claim 2, wherein said co-processor comprising:

a program counter for holding an address of an instruction which is currently processed and belongs to said second instruction set; and

a hardware resource for carrying out a process for updating said program counter among processes which take place in the course of execution of said specific instruction.

10. A microprocessor system according to claim 2, wherein said co-processor comprises a status register for holding information indicative of a need of said notification and wherein said main processor periodically accesses said status register to recognize, from content of said status register, that said co-processor has encountered said specific instruction to thereby execute said specific instruction.

11. A microprocessor system according to claim 4, wherein said main processor further comprises an interrupt request reception circuit for encoding said dedicated interrupt vectors sent from said co-processor to specify an interrupt handler which corresponds to said specific instruction to be processed.

12. A microprocessor system according to claim 2, wherein said co-processor further comprises an instruction queue for holding a fetched instruction which belongs to said second instruction set and wherein said main processor refers to said instruction queue of said co-processor to specify an interrupt handler which corresponds to said specific instruction to be executed.

A2

13. A microprocessor system according to claim 1, wherein said co-processor has a stack architecture.

14. A microprocessor system according to claim 13 further comprising a stack memory provided outside said co-processor, wherein said co-processor further comprises a stack-top register for holding a predetermined number of top data of stack data.

15. A microprocessor system according to claim 14, wherein said co-processor further comprises a cache memory provided between said stack memory and said stack-top register for caching a part of data held in said stack memory.

16. A microprocessor system according to claim 14, wherein said co-processor detects a predetermined instruction for which stack data needs to be manipulated over said stack-top register and said stack memory, whereupon said co-processor moves contents of said stack-top register to said stack memory and thereafter requests said main processor to execute said predetermined instruction, said main processor referring to contents of said stack memory, to which said contents of said stack-top register have been moved, to thereby execute said predetermined instruction.

17. A microprocessor system according to claim 1 comprising a plurality of co-processors in correspondence with a plurality of processes described in a program.

18. A microprocessor system according to claim 1 further comprising a program memory in which instructions belonging to said second instruction set are contained, wherein said co-processor further comprises:

a program counter for holding an address of an instruction that is currently processed and belongs to said second instruction set;

an instruction queue for holding instructions which belong to said second instruction set; and

an instruction fetch circuit for fetching an instruction belonging to said second instruction set from said program memory using a value contained in said program counter as its address and for setting the fetched instruction to said instruction queue.

Add A2